USEFUL LINKS

* Angular tutorial through w3schools - <https://www.w3schools.com/angular/>
* Angular directives reference - <https://www.w3schools.com/angular/angular_ref_directives.asp>

INSTALLING IN YOUR APP/PROJECT

ABOUT ANGULAR

* AngularJS is a JavaScript framework/library that can be added to an html page. It extends HTML attributes with “directives” and binds data to html with “expressions”.
* Angular models define angularjs apps, the controllers control it. The **ng-app** directive defines the application, the ng-controller directive defines the controller.

FILE/FOLDER FORMAT

WORKFLOW

**THE BASICS**

* AngularJS extends HTML with **ng-directives**, some common and important ones are: **ng-app**, which goes at the top of index.html and defines an angularJS application, **ng-model** which binds the value of html controls (input, select, textarea) to application data, and **ng-bind** which binds application data to the HTML view. All directives start with “**ng-**”
* Angular expressions are written within html code in double curly brackets. Ex: if you have in your html page: **<h1>{{ 5+5 }}</h1>**, then 10 would be written as a header 1. But expressions can also be written inside a directive, ex: **ng-bind=“expression”**
* **JavaScript Expressions VS AngularJS Expressions:** Both can contain literals, operators, and variables. But AngularJS can’t be written inside HTML, nor can you use conditionsl, loops, and exceptions with angularjs.
* **ANGULARJS CONTROLLERS:** Are javascript objects created by a standard javascript object constructor. You define their application or scope with **ng-controller**, meaning that if I want a partial to use a certain controller, I would wrap the entire partial in say a div, and write **<div ng-controller= ‘controller\_name\_here’>**

THE DIFFERENCE BETWEEN USING $SCOPE AND NOT WITHIN THE CONTROLLERS

**ANGULARJS DIRECTIVE BASICS AND EXAMPLES**

* **NG-APP:** is where you define the root element of the angularjs application. This directive will “auto-bootstrap” (automatically initialize) the application when a web page is loaded. This can go at the start of your body but is better off at the start of your index.html file, ex: **<html ng-app='myApp'>**
* **NG-INIT:**is where you can basically declare variables or gives things values. For example, if I wrote **<div ng-init=“myCol= ‘lightblue’ “>**, then in another line wrote **<input style= “background-color: {{myCol}}” ….>**, then it would read that as “background-color: ‘lightblue’”. Another example, you could declare two variables by doing **<div ng-init= “quantity=1; cost=5; name=‘John’”>**, and use either variable as you want. Every directive, ng-init included, is like class in the sense that you put them in tags, ex: <div directive\_here>, <input directive\_here>, <p directive\_here>. You could also create objects or arrays using ng-init.
* **NG-BIND:** like it sounds, literally binds a value to something. If I wrote: **<p> The third result is <span ng-bind=“16”></span></p>**, then it would translate to “The third result is 16”.
* **NG-MODEL:** This directive allows you to bind the value of an input field to a variable created in angularJS, for example, if in my controller I have **this.name = ‘Jack’** (not using $scope in this example), and then had an input like this: **Name: <input ng-model=“controllerName.name”>**, then this would show up on the page as an input with “Jack” written in it to start. This is a two way binding however, meaning that if I click that input and typed to change it to “Jack swanson”, then the value of “this.name” in the controller would also change, until I reloaded the page/partial, they are linked.
* **NG-REPEAT:** is a directive used to repeat an html element, similar to a for loop in javascript. For example, if I wrote **<div ng-init= “names=[‘Jani’, ‘Hege’, ‘Kai’]”>**, then wrote:

**<ul>**

**<li ng-repeat=“x in names”>**

**{{x}}**

**</li>**

**</ul>**

* **STILL NG-REPEAT:** Then it would show each index in the names array as a list item. You could also use this on an array of objects (like data you get from mongoDB sometimes). For example, if names was **[ {name: ‘Jani’, country: ‘Norway’}, {name: ‘Hege’, country: ‘Sweden’}, {name: ‘Kai’, country: ‘Denmark’} ]**. Then you would write the repeat code as:

**<ul>**

**<li ng-repeat=“x in names”>**

**{{ x.name + ‘, ’ + x.country }}**

**</li>**

**</ul>**

* **CREATING DIRECTIVES**: I have yet find a need to create a new directive but you can with the **.directive** function, see how in this url: <https://www.w3schools.com/angular/angular_directives.asp>, in the middle of the page is where it shows it.